

AMENDMENTS TO THE CLAIMS

1.-32. (Canceled)

33. (Previously Presented) A method of implementing per-instance attributes, the method comprising computer-implemented steps of:
- instantiating a first instance of a class and a second instance of said class, wherein
- said class includes a single-valued property attribute;
- wherein both said first instance of said class and said second instance of said class
- include said single-valued property attribute;
- assigning to said single-valued property attribute of said first instance a first value
- that associates a first set of one or more attributes with said first instance;
- assigning to said single-valued property attribute of said second instance a second
- value that associates a second set of one or more attributes with said second
- instance;
- wherein said first set of one or more attributes is different than said second set of one
- or more attributes;
- wherein said first set of one or more attributes and said second set of one or more
- attributes are not included in said class or any superclass of said class;
- wherein assigning said first value to said single-valued property attribute of said first
- instance does not cause any of said first set of one or more attributes to
- become attributes of said class or of any instance of said class other than said
- first instance;
- wherein assigning said second value to said single-valued property attribute of said

second instance does not cause any of said second set of one or more attributes to become attributes of said class or of any instance of said class other than said second instance;

wherein said steps of said method are performed by one or more computing devices.

34. (Previously Presented) The method of Claim 33, wherein:
- a first property class contains said first set of one or more attributes, wherein said first value references an instance of said first property class; and
- a second property class contains said second set of one or more attributes, wherein said second value references an instance of said second property class; and
- said first property class is different than said second property class.
35. (Previously Presented) The method of Claim 33, wherein:
- said first value references a first instance of a property bundle class, wherein one or more instances of a first set of one or more property classes include pointers to said first instance of said property bundle class, wherein said first set of one or more property classes include said first set of one or more attributes;
- said second value references a second instance of said property bundle class, wherein one or more instances of a second set of one or more property classes include pointers to said second instance of said property bundle class, wherein said second set of one or more property classes include said second set of one or more attributes; and
- said first set of one or more property classes are different than said second set of one or more property classes.

36. (Previously Presented) The method of Claim 33, further comprising the steps of:
- assigning a first set of one or more keys to said first set of one or more attributes, wherein said first set of one or more keys respectively identifies said first set of one or more attributes; and
- assigning a second set of one or more keys to said second set of one or more attributes, wherein said second set of one or more keys respectively identifies said second set of one or more attributes.
37. (Currently Amended) The method of Claim 36, wherein:
- assigning said first set of one or more keys comprises receiving, from a user, first data that defines said first set of one or more keys; and
- assigning said second set of one or more keys comprises receiving, from said user, second data that defines said second set of one or more keys.
38. (Currently Amended) The method of Claim 36, wherein:
- assigning said first set of one or more keys comprises invoking a method for each of said first set of one or more attributes.
39. (Previously Presented) The method of Claim 33, wherein said class represents a file type in an extensible file system, and wherein:
- said first instance of said class represents a first file of said file type, wherein said first file is associated, through the first value, with said first set of one or more attributes that do not belong to said file type; and

said second instance of said class represents a second file of said file type, wherein
said second file is associated, through the second value, with said second set
of one or more attributes that do not belong to said file type.

40. (Previously Presented) A method of implementing per-instance attributes, the
method comprising computer-implemented steps of:

instantiating an instance of a class, wherein said class defines a first set of one or
more attributes;

including, in the structure of said instance of said class, a table for storing data
associated with a second set of one or more attributes, wherein, for each
particular attribute of said second set of one or more attributes, said table
includes a particular entry that comprises:

a first field for storing an identifier of said particular attribute; and
a second field for storing one or more data values of said particular attribute;

wherein said first set of one or more attributes defined in said class do not include
said second set of one or more attributes;

wherein said second set of one or more attributes do not belong to said class or any
superclass of said class;

wherein said steps of said method are performed by one or more computing devices.

41. (Previously Presented) The method of Claim 40, wherein said identifier is an
attribute name of said particular attribute.

42. (Previously Presented) The method of Claim 40, wherein identifiers stored in

said first field are designated as keys for locating entries in said table, and wherein said method further comprises steps of:

receiving a data value of a specific attribute of said second set of one or more attributes and a specific identifier that identifies said specific attribute; using said specific identifier to locate in said table a specific entry that stores said specific attribute; and without performing datatype checking on said data value, storing said data value in said specific entry.

43. (Previously Presented) The method of Claim 40, wherein said table is a hash table, and wherein said method further comprises steps of:
- receiving input that identifies a specific attribute of said second set of one or more attributes;
- hashing said input to generate a hash key; and
- using said hash key to locate in said hash table a specific entry that stores said specific attribute.
44. (Previously Presented) The method of Claim 40, further comprising a step of adding a specific attribute to said second set of one or more attributes in response to an invocation of a method defined in said class.
45. (Previously Presented) The method of Claim 44, wherein:
- said invocation of said method includes a specific identifier by which said specific attribute will be identified; and

adding said specific attribute includes storing said specific identifier in said first field
of a specific entry in said table.

46-58. (Canceled)

59. (New) A computer-readable storage medium storing one or more sequences of executable instructions for implementing per-instance attributes, which instructions, when executed by one or more processors, cause performance of steps comprising: instantiating a first instance of a class and a second instance of said class, wherein said class includes a single-valued property attribute; wherein both said first instance of said class and said second instance of said class include said single-valued property attribute; assigning to said single-valued property attribute of said first instance a first value that associates a first set of one or more attributes with said first instance; assigning to said single-valued property attribute of said second instance a second value that associates a second set of one or more attributes with said second instance; wherein said first set of one or more attributes is different than said second set of one or more attributes; wherein said first set of one or more attributes and said second set of one or more attributes are not included in said class or any superclass of said class; wherein assigning said first value to said single-valued property attribute of said first instance does not cause any of said first set of one or more attributes to

become attributes of said class or of any instance of said class other than said first instance; and

wherein assigning said second value to said single-valued property attribute of said second instance does not cause any of said second set of one or more attributes to become attributes of said class or of any instance of said class other than said second instance.

60. (New) The computer-readable storage medium of Claim 59, wherein:

a first property class contains said first set of one or more attributes, wherein said first value references an instance of said first property class; and
a second property class contains said second set of one or more attributes, wherein said second value references an instance of said second property class; and
said first property class is different than said second property class.

61. (New) The computer-readable storage medium of Claim 59, wherein:

said first value references a first instance of a property bundle class, wherein one or more instances of a first set of one or more property classes include pointers to said first instance of said property bundle class, wherein said first set of one or more property classes include said first set of one or more attributes;
said second value references a second instance of said property bundle class, wherein one or more instances of a second set of one or more property classes include pointers to said second instance of said property bundle class, wherein said second set of one or more property classes include said second set of one or more attributes; and

said first set of one or more property classes are different than said second set of one or more property classes.

62. (New) The computer-readable storage medium of Claim 59, wherein the one or more sequences of executable instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of:
assigning a first set of one or more keys to said first set of one or more attributes,
wherein said first set of one or more keys respectively identifies said first set of one or more attributes; and
assigning a second set of one or more keys to said second set of one or more attributes, wherein said second set of one or more keys respectively identifies said second set of one or more attributes.
63. (New) The computer-readable storage medium of Claim 62, wherein:
the instructions that cause performance of the step of assigning said first set of one or more keys comprise instructions which, when executed by the one or more processors, cause performance of the step of receiving, from a user, first data that defines said first set of one or more keys; and
the instructions that cause performance of the step of assigning said second set of one or more keys comprise instructions which, when executed by the one or more processors, cause performance of the step of receiving, from said user, second data that defines said second set of one or more keys.
64. (New) The computer-readable storage medium of Claim 62, wherein:

the instructions that cause performance of the step of assigning said first set of one or more keys comprise instructions which, when executed by the one or more processors, cause performance of the step of invoking a method for each of said first set of one or more attributes.

65. (New) The computer-readable storage medium of Claim 59, wherein said class represents a file type in an extensible file system, and wherein:
said first instance of said class represents a first file of said file type, wherein said first file is associated, through the first value, with said first set of one or more attributes that do not belong to said file type; and
said second instance of said class represents a second file of said file type, wherein said second file is associated, through the second value, with said second set of one or more attributes that do not belong to said file type.
66. (New) A computer-readable storage medium storing one or more sequences of executable instructions for implementing per-instance attributes, which instructions, when executed by one or more processors, cause performance of steps comprising:
instantiating an instance of a class, wherein said class defines a first set of one or more attributes;
including, in the structure of said instance of said class, a table for storing data associated with a second set of one or more attributes, wherein, for each particular attribute of said second set of one or more attributes, said table includes a particular entry that comprises:
a first field for storing an identifier of said particular attribute; and

a second field for storing one or more data values of said particular attribute;
wherein said first set of one or more attributes defined in said class do not include
said second set of one or more attributes; and
wherein said second set of one or more attributes do not belong to said class or any
superclass of said class.

67. (New) The computer-readable storage medium of Claim 66, wherein said identifier is an attribute name of said particular attribute.
68. (New) The computer-readable storage medium of Claim 66, wherein identifiers stored in said first field are designated as keys for locating entries in said table, and wherein the one or more sequences of executable instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of:
receiving a data value of a specific attribute of said second set of one or more attributes and a specific identifier that identifies said specific attribute;
using said specific identifier to locate in said table a specific entry that stores said specific attribute; and
without performing datatype checking on said data value, storing said data value in said specific entry.
69. (New) The computer-readable storage medium of Claim 66, wherein said table is a hash table, and wherein the one or more sequences of executable instructions further

comprise instructions which, when executed by the one or more processors, cause performance of the steps of:

receiving input that identifies a specific attribute of said second set of one or more attributes;
hashing said input to generate a hash key; and
using said hash key to locate in said hash table a specific entry that stores said specific attribute.

70. (New) The computer-readable storage medium of Claim 66, wherein the one or more sequences of executable instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of adding a specific attribute to said second set of one or more attributes in response to an invocation of a method defined in said class.
71. (New) The computer-readable storage medium of Claim 70, wherein:
said invocation of said method includes a specific identifier by which said specific attribute will be identified; and
the instructions that cause performance of the step of adding said specific attribute include instructions which, when executed by the one or more processors, cause performance of the step of storing said specific identifier in said first field of a specific entry in said table.